VACUUM

When records are deleted from a table through an SQL DELETE, the deleted records are referred to as "dead tuples". They are not removed from the db and the space they reside in is NOT MARKED AS BEING AVAILABLE FOR RE-USE. The same situation occurs for updated records. The VACUUM command must be run to make the "dead tuple" space available for reuse.

Running the vacuum command (without the "full" option) marks previously deleted (or updated) records as being available for re-use within the table. It does not lock the table being vacuumed. Running vacuum with the "analyze" option reads the records in the tables and generates statistics used by the queries. This information is stored in the pg_statistics table.

The SwEG's policy on running vacuum is to have the postgres cron submit vacuum runs for each of the standard AWIPS databases. This will prevent the possibility of multiple vacuum runs executing at the same time which can cause a slowdown in the server. A vacuum/analyze of the IHFS db has been scheduled to be submitted from the fxa cron every 4 hours.

A "vacuum full" attempts to remove deleted or updated records from the tables to make the space reusable by other tables. It physically reorders the tables. While "vacuum full" is running, an exclusive lock is placed on the table being vacuumed. This locks the table for both reads and writes. A "vacuum full" run is NOT necessary to be run on a regular basis. The SwEG is discussing the set up of an ITO Alarm for the case of a database getting very large where a run of "vacuum full" would shrink the database back to normal size.

"vacuum" and "vacuum full" can be run for an entire database or for individual tables. See Section 21.1.1 of the PostgreSQL 7.4 Documentation entitled "Recovering disk space" for information on strategies for running VACUUM.

Explanation of Vacuum Log Output

A log file is generated by each vacuum run. These logs are being written to the $LOG_DIR directory which normally points to the /data/logs/fxa directory. OHD and its PostgreSQL consultant will be monitoring these log files to watch for problems such as a slow increase in size of the db over time. We also hope to glean information from the logs which will be used to tweak the configuration parameters.

The last few lines of output from running a vacuum on an OHD db look as follows:

```
INFO: free space map: 901 relations, 6879 pages stored; 74608 total
pages needed
DETAIL: allocated FSM size: 1000 relations + 20000 pages = 178 kB
shared memory.
VACUUM
```

```
On the "INFO" line, "901 relations" signifies that a total of 901
tables currently exist across all databases on the server.

In the "DETAIL" line, FSM is an acronym for Free Space Map.  This line
shows that space has been allocated for a maximum of 1000 tables and
20000 pages for all postgresql databases on the server.
```

Vacuum Script

The following script is submitted via the postgres cron on dx1 at all AWIPS sites to vacuum the PostgreSQL databases. The filename is /awips/ops/bin/vacuum_pgdb.

```
#!/bin/bash
# Sccsid_vacuum_pgdb="@(#)vacuum_pgdb.sh 1.0.2 04/27/2005 13:03:05";
#
# NAME
#   vacuum_pgdb - Vacuum a postgres database
#
# SYNOPSIS
#   vacuum_db -d db_name [-z]
#
# DESCRIPTION
#   This script calls the vacuumdb executable to vacuum AWIPS
databases.
#   It will normally be run from the "postgres admin user" cron.
#
#   The command line of the script is
#
#       vacuum_db -d db_name,... [-z]
#
#       db_name = the name of the database to be vacuumed
#
#   or
#       vacuum_db -a -x db_name,... [-z]
#
#       db_name = the name of the database to exclude from the vacuum
#
#   The "-z" option is optional.  If it appears on the command line,
#   then the vacuum will also perform an "analyze".
#
#   This script logs output to /data/fxa/logs/vacuum_${DBNAME}_MMDD.
#
# HISTORY
#   4/08/2005   Original Version (Paul Tilles)
#   4/19/2005   Updates for environment vars, command line options
checking,
#               log file name (append to daily log instead of one log
per
#                   vacuum), and exit code.
#
##############################################################################

USAGE="Usage: $0 -a|-d dbname_list [-x exclude_list] [-z]"
FXA_HOME=${FXA_HOME:-~fxa}
unset ANALYZE
```

```
VACUUM="vacuum"
VACUUM_ALL=0
unset DBNAME_ARRAY
unset EXCLUDE_ARRAY

# Read the command line args
while getopts :ad:x:z opt ; do
    case $opt in
        a ) VACUUM_ALL=1
            ;;
        d ) DBNAME_ARRAY=( ${OPTARG//,/ } )
            ;;
      x ) EXCLUDE_ARRAY=( ${OPTARG//,/ } )
          ;;
        z ) ANALYZE="--analyze"
            VACUUM="vacuum analyze"
            ;;
        * ) echo $USAGE;
            exit 1
            ;;
    esac
done

if [ -z "${DBNAME_ARRAY[*]}" -a $VACUUM_ALL -eq 0 ] ; then
    echo $USAGE
    exit 1
fi

# Source the AWIPS and PostgreSQL environments
. $FXA_HOME/readenv.sh
. postgresenv.sh

PSQL_BIN_DIR=$PG_INSTALL/bin

if [ $VACUUM_ALL -ne 0 ] ; then
    DBNAME_ARRAY=( $($PG_INSTALL/bin/psql -U postgres --list --tuples-
only \
      | while read _DBNAME _JUNK; do \
      if [ "$_DBNAME" != "template0" -a "$_DBNAME" != "template1" ] ;
then \
      echo $_DBNAME; fi; done) )
fi

for EXCLUDE in ${EXCLUDE_ARRAY[*]} ; do
    let "I = 0"
    while [ ! -z "${DBNAME_ARRAY[$I]}" ] ; do
      if [ "${DBNAME_ARRAY[$I]}" = "$EXCLUDE" ] ; then
          unset DBNAME_ARRAY[$I]
      fi
      let "I = $I + 1"
    done
done

if [ -z "${DBNAME_ARRAY[*]}" ] ; then
    echo "Nothing to vacuum!" > /dev/stderr
    exit 1
fi
```

```
#########################################################################
# Run vacuumdb
# Write database name, begin time and end time to log

EXIT=0
for DBNAME in ${DBNAME_ARRAY[*]} ; do

    LOGFILE=$LOG_DIR/vacuum_${DBNAME}_$(date -u +%m%d_%H%M)

# We should not need to do this, all databases should be owned by
pguser
    USERNAME=$($PG_INSTALL/bin/psql -U postgres --list --tuples-only |
\
      while read _DBNAME _DELIM _USERNAME _JUNK ; do \
      if [ "$_DBNAME" = "$DBNAME" ] ; then echo $_USERNAME; fi; \
      done)

    echo $(date +"%b %d %T") BEGIN $VACUUM $DBNAME as $USERNAME >>
$LOGFILE

    $PSQL_BIN_DIR/vacuumdb -v $ANALYZE -U $USERNAME $DBNAME >> $LOGFILE
2>&1
    RETURN=$?

    if [ $RETURN -ne 0 ] ; then
      EXIT=$RETURN
    fi

    DTZ=`date -u +%T`
    echo $(date +"%b %d %T") END $VACUUM $DBNAME EXIT_CODE=$RETURN >>
$LOGFILE

done

exit $EXIT
```

Submitting from cron

The vacuum_pgdb script is submitted via the cron on dx1 as follows:

```
# DX1apps postgres crontab
# This is the cluster-managed crontab for postgres
00 00,04,08,12,16,20 * * * postgres . ${FXA_HOME:=~fxa}/readenv.sh;
/awips/ops/bin/vacuum_pgdb -d hd_ob6$(echo $FXA_INGEST_SITE | tr "[A-
Z]" "[a-z]") -z

05 02 * * * postgres /awips/ops/bin/vacuum_pgdb -d hmdb
05 05 * * * postgres /awips/ops/bin/vacuum_pgdb -d hmdb -z
05 03 * * * postgres /awips/ops/bin/vacuum_pgdb -d fxatext
05 07 * * * postgres /awips/ops/bin/vacuum_pgdb -d fxatext -z
05 09 * * * postgres . ${FXA_HOME:=~fxa}/readenv.sh;
/awips/ops/bin/vacuum_pgdb -a -x fxatext,hd_ob6$(echo $FXA_INGEST_SITE
| tr "[A-Z]" "[a-z]") –z
```

The above statements are located in files in the /etc/cron.d directory.

Other Info

The following site offers an interesting discussion concerning vacuuming:

http://pgsqld.active-venture.com/routine-vacuuming.html